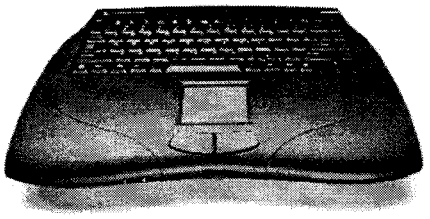
LOADSTAR LETTER #61

Commodore Clone Makers Release A Little More Hype And Info

By Jeff Jones. This is all I could find at The Web Computer web site this month. The site was still heavily under construction until recently when it was almost completely revamped. Here's the latest release:

After the successful introduction of the Web.it Internet Computer on the CeBIT Home exhibition in Hanover, Germany, now Web Computers



International is proud to present the new generation Commodore 64 to the UK audience.

This user-friendly all-in-one keyboard computer that you can connect to your TV will be introduced during the Live 98 event at Earls Court, London from 24-27 September on Stand K18.

Stylishly designed all-in-one computer CeBIT Hanover (Germany), 26 August 1998 -- Web Computers International today launched the Web.it Internet computer. An easy-to-use home computer for all the family, the Web.it will be available at electrical shops and department stores in Germany, the UK and France from the end of September. In October the United States and Benelux will follow.

Thanks to its stylish design, Web.it is a boon to any living room. And for anyone who doesn't want to have an expensive office-style computer at home, it's the ideal way to surf the Internet and to send and receive e-mail. Web.it is in many ways reminiscent of its renowned predecessor, the Commodore C64. It's just about as small and, like the C64, Web.it can be linked directly to a television set. What's more, Web.it is supplied with an impressive package of software: the popular Netscape program for Internet surfing and e-mail, a Lotus word processor, the celebrated Lotus 1-2-3 spreadsheet program, Lotus Organizer and -- last but not least -- Commodore 64 emulator, which enables it to play C64 games.

The user can start any of these programs using the hot-keys which form the top row of the keyboard. Thanks to these handy keys, switching between programs is a user-friendly procedure. The complete, all-in-one Web.it living-room computer retails just below US $400. This price includes every imaginable facility: telephone connector and modem, keyboard, touchpad, penpad and ports for floppies, a printer, a PC card, MIDI, a separate monitor, a projector, and so on. Web.it is also easy to enhance with an optional infrared remote control unit.

IBM and Lotus participation: Web. it was devised and developed in close co-operation with IBM and Lotus, which handled its software aspects. The development partners opted to incorporate embedded software, so that the user can operate all the functions of Web.it instantly and at the touch of a single button. The ease-of-use and low price of Web.it make the Internet accessible for all. The long-term global co-operation between Web Computers International, IBM and Lotus demonstrates their high expectations that Web.it can bridge the business and consumer markets for the net. As an Internet computer, Web.it offers consumers the opportunity to participate in the e-commerce developments taking place on the net, including electronic shopping.

"We chose to work with IBM and Lotus because of their vast experience in the field of thin clients and their technological expertise in Web computers and set-top boxes," says Gerard Lindhout, Director of Web Computers International.

Web Computers International is based in the Netherlands Antilles and has a subsidiary in Antwerp, Belgium. The company was founded in 1998 with the objective of developing a new home-computing standard. Web Computers International stands out by responding flexibly to market demand and integrating technology from leading market players into a unique and affordable product.

For more information, contact: Web Computers International AEC Antwerp Branch Lange Lozanastraat 176-182 B-2018 Antwerp, Belgium E-mail: info@webcomputers.net Website: http://www.webcomputers.net

Commodore Stereo--Sort of...

By Scott Eggleston. Why is it that all of our monitors seem to have their speakers pointing upward or to the side? None of them point directly at the user, unlike all modern stereo monitors sold today.

When I picked up my 1084 a couple of years ago, I didn't even notice the direction of the speaker until I put it in a desk with a piece of wood about four inches from it. It muffled the sound horribly, and made me want to find a solution.

The 17xx series of monitors will often encounter the same problem. While on the top rather than the side, the sound is instantly muffled the minute you set anything on top of your monitor.

The solution is to point the sound directly at you, via a single or pair of external speakers. While you can buy these new from any electronics/computer parts catalog, I turned to the greatest resource of Commodore stuff--the thrift store.

As PC speaker systems turn more into home theater systems, older, smaller speakers are being discarded. I began keeping my eyes open, and finally found

jeff@LOADSTAR.com
US MAIL: ATTN. Jeff Jones
J & F Publishing
P.O. Box 30008 Shreveport
LA 71130-0008
Phone: 318/221-8718,
Fax: 318/221-8870

a pair of Labtec CS-150 amplified speakers. I snagged them for 5 bucks, ran home, and gave them a test. They worked perfect, with only a little evidence of grime that they had been used before. They are a nice cream color that matches my 128 system rather well.

One nice thing about the 1084 is the 1/8" out jack just below the speaker toward the back of the monitor. It was easy to plug the 1/8" speaker plug directly into the monitor to test it. Since you are plugging right into the monitor, the speaker volume is still controlled by the monitor's volume knob, which is a nice touch.

Due to our computers being mono and not stereo, sound only came through one of the speakers. No problem. Radio Shack carries a 1/8" mono-to-stereo adapter plug (part #274-338c, $2), which will send the sound to both speakers. Granted, this isn't stereo, just split mono.

For true stereo, you'll need a Stereo SID cart from CMD, or make your own extra SID hack. This will give you an extra 3 voices to play with, which can be sent to the other speaker.

If you don't have a 1084, you're going to have to buy or make some adapters to take the 1/8" stereo plug. For mono you'll need an RCA female to 1/8" phone plug female, if you're using standard monitor cables. For stereo, you'll probably have to make the zany thing, since you'll need a combination of connectors depending on your setup.

The point of all this is now you can place the output of your sound in the most pleasing locations), not being hampered by the teeny one built-in to your monitor. I recommend the amplified versions versus the non-amp, as they offer cool features such as bass/treble boost, offering crisper, cleaner sound. Boot up a demo, be amazed!

Robin's comp.sys. cbm Rantings

RCKCLMR13 wrote:
Which one is better? To have JiffyDOS on the C64 or on the disk drive?

Robin: If you could only have one, definitely go with it in your C64. I was pretty thick in (at least) one of my other postings - I could only think of two features JiffyDOS had besides the disk speed up...

Here's a few more:

- Screen dump built-in
- Ctrl-D to flip through available disk devices
- Disk to disk copy
- Non-destructive SEQ file reader
- Non-destructive BASIC program lister
- Built-in DOS wedge
- lock/unlock files
- Change interleave gap
- un-new basic programs
- disable drive rattle

There may be more... but the main point of JiffyDOS is to speed up your drives, so really you should get the ROMS for both drive and computer.

RCKCLMR13 wrote: ***Here and there, I've seen many different disk drives. Unfortunately, I have no idea if all of them are compatible with the C64. Therefore, I am gonna list all of them and ask for your opinions on which one you think is better for the C64 among them.***

A. CBM 1541-II
B. CBM 1541C
C. CMD FD-2000 (800K and 1.6MB)
D. 1571

Robin: For a C64, the 1541-II/ C/1571 are all pretty much the same, for general use, anyway. I figure everyone needs at least one of these drives - and they are pretty darn slow if you don't have some kind of speed up. My favorite drive is the FD-2000. I use it whenever I can - but there is a lot of software out there that assumes you have a 1541, and won't work properly with the FD-2000.

The FD-2000 is a fantastic second drive. In fact, it's become my first drive, while the 1571 is there when I need it. It has JiffyDOS built in, and is even fairly fast without JiffyDOS in the computer. It has real sub-directories, partitions if you want them, and 6400 blocks free sure is nice :)

Joe Forster/STA wrote: ***Hi Greg, the answer is simple: Cause it's*** **(purchasing a SuperCPU etc. – editor)** ***damn expensive. For that price, you could get a 386 or something with a lot more computing power, you could even run Arachne, a DOS-based graphical Web browser on it which is, in my opinion, almost as fast as Netscape.***

Robin: This is a common argument, and certainly has truth to it. But I don't think it's this simple. Many people have spent a lot of time and money over the last 15 years to build up their C= systems - they don't want to have to learn a whole new computer system. They want to use the same apps they've been using all along. But the extra power is something useful, something handy - be it in GEOS, or while running your own home-brewed programs (especially in BASIC). Whenever I'm enjoying my latest issue of Loadstar or Driven, I always have the SuperCPU on - I think it's great how fast everything pops up. GEOS/ Wheels is amazing at 20mhz.

I'm a programmer, and a user - and I think the SuperCPU is great. Sure, I've got other newer machines, but I still think the SuperCPU is a worthwhile thing.

Joe Forster/STA: *expensive. It's a very nice idea to upgrade your Commodore machines but not for their prices... That's my opinion.*

Robin: It depends what way you look at it. A 2000% speedup for $200? People in the PC world would freak out if such a device were made. And your SuperCPU isn't going to lose its value the way a more modern machine will. I'm sure if I posted my SuperCPU for sale here for $100, I'd sell it within the day - not bad, my system only "devaluing" $100 in two years... tell that to my 6 month old K6/233 - it's already worth $200 less, even if it was brand new!

Subject: Re: TCP/IP, HTML is it feasible on the C64? (The thread had been discussing a C64 based web browser –editor)

Someone wrote: *If made specifically for WHEELS then all the better. The User would HAVE to have some sort of RAM expansion, and if s/ he doesn't own Wheels or extra Ram, what a perfect incentive to invest in some.*

Robin: Well - I don't bash this idea, I actually think it's got a lot of merit. Here's my "proof" :)

Question: Out of all the apps on the c64, which one looks/feels most like a web browser?

Answer: geoWrite. I'm not saying it *acts* like a web browser - but when I'm scrolling through a geoWrite document that has graphics and multiple fonts and bold and underline etc in it - boy, I can easily imagine I'm looking at a web page.

I haven't done much GEOS programming, but I do have all the tools and books necessary to do it - while reading through, it seems a lot of the work necessary to making a web browser has already been done for you in the GEOS kernal.

Of course someone could write a dedicated web browser for the 64 that would be more efficient. But many people like to work within the GEOS environment, and having all the other GEOS apps available at your disposal quickly/instantly sure would be nice.

More On JiffyDOS

Robin: ...you're completely right as far as the speed benefits of JiffyDOS go. However, it's not pointless and worthless to have JiffyDOS ROMs in your computer - at least you get the programmable function key (especially that F1 directory key :)

Matthew Montchalin: *Nothing new there. People have written directory listing routines for the c-64 for ages.*

Robin: Of course! My point is that it's *built in* - I don't have to go burning my own ROMs from someone else, or writing my own, and I *don't* have to go fumbling around to load some program when all I want to do is quickly display a directory.

I certainly hope you can understand the benefits of having a built-in DOS wedge/shell. I do, and most everyone else here understands that. That was my point.

Matthew Montchalin: Now, what about filenames that have embedded carriage returns or null bytes in them? Are they displayed as reverse m's and @'s, or do they foul up the JiffyDOS display? (I'll stick with my own routines, thank you.)

Robin: Just for the heck of it, I went and put a null byte (I assume you mean a byte with a value of 0) in a file name. I did a directory with my Super Snapshot active. Sure enough - the directory was "fouled up". I turned my SuperCPU on (With the JiffyDOS switch on the SuperCPU unit set to "enable") and did a directory - sure enough, messed up there too. Then I tried doing a directory on my stock 128D - in both 64 and 128 modes, it messed up there too.

So, JiffyDOS does exactly the same thing that a stock 64/128 does - what's your argument? That we should all use your kernal replacement, or we should all code our own kernal?

JiffyDOS fixed the CAPS-LOCK Q bug that some 128s apparently suffer from - how about the Save with Replace bug? JiffyDOS fixes that as well. Anyone else?

Cameron Kaiser: This is my general complaint on the subject, and is not intended to be a slam on you or on CMD, who has done a very commendable engineering job on their peripherals, not to mention staying in the market. But it looks awfully like people are jumping ship from the stock machine in a hurry.

Robin: I bought a C64 in 1983 with *no* storage device. I bought a Datasette within about 3 months and thought it was great being able to save the programs I had written. About 2 years later I could finally afford a 1541, and grabbed it happily. That's two "upgrades" for me, and yet that's what most people mean by a "stock" C64 system.

Each of those upgrades I made allowed me to work quicker, and generally have more fun with my C64. I had more options available to me. That's exactly the same thing I'm doing today, as I add a REU, a SuperCPU and a FD-2000 to my system.

BBGRam/ geoRAM Programming

By Robin Harbron. I remember reading about the geoRAM quite a few years ago, when it first appeared on the market. The good news was the price – it was cheaper than a REU (Ram Expansion Unit) of the same capacity. The bad news was the claim that it would only work in GEOS. I wasn't a GEOS user at the time, so I forgot about the unit.

A number of years later, I was searching the Internet for used RAM expansion for my C64. I managed to find both a 512k REU, and a 2MB BBGRam. I was told that the BBGRam was geoRAM compatible – the price wasn't bad, so I picked both units up.

The BBGRam and geoRAM actually will work apart from GEOS. They're just not useful for most people except in GEOS because there are no programs that support the devices – at least none that I know of.

I don't own a geoRAM, but I've been told that programming for one is the same as the other. I'll just refer to the device as a BBGRam from now on, as that's what I have been using.

While programming for this device, I realized there is a bit more good news: it's much easier to program than a REU. While a REU has 10 or so registers, the BBGRam has only 2. Locations $DFFE and $DFFF allow you to select which 256 byte page of RAM within the device you want to work with. My 2MB unit has 8192 of these pages available – the 1MB and 512K versions have 4096 and 2048 pages, respectively.

The selected page of RAM appears at locations $DE00-$DEFF – you can read and write to it just as if it were regular RAM inside the C64. It works fine in BASIC, as well as machine code.

There is a downside to this, however. A REU can transfer information at an incredible rate – as fast as 1MB per second! The BBGRam, on the other hand, can only work as fast as the C64 can move information internally – at best, about 8 times slower. Additionally, the extra overhead involved with only being able to look at 256 bytes at a time will cause things to run even slower, in certain situations. And finally, other devices that use the $DExx or $DFxx memory range are incompatible with the BBGRam. This rules out most fast load and utility cartridges, such as the Super Snapshot and Action Replay. Similarly, an REU cannot be used with the BBGRam, nor can a Turbo232 or SwiftLink be used in their default mappings.

The SuperCPU almost equalizes the BBGRam and REU. The SuperCPU can only read and write memory internally at 20Mhz – whenever it tries to communicate with the "outside world" (such as the REU or BBGRam) more than once per 1mhz cycle, it has to

slow down to 1mhz. The end result? The SuperCPU will grab one byte per 1mhz cycle from either device – the REU getting a slight edge whenever the number of bytes needed is more than one page's worth, and the BBGRam possibly being slightly faster for smaller transfers, due to less overhead in the initial setup.

The setup of the two registers is slightly strange. I'm sure there's a good reason for the way the designers made this unit, but I can't figure it out. $DFFE holds the low 6 bits (in bits 0-5) of the page number you want to access, while $DFFF holds the high 5, 6 or 7 bits (in bits 0-4, 0-5 or 0-6), depending on whether you have the 512k, 1MB or 2MB model.

Say I want to access location 510927 in my BBGRam. Let's turn that into hex: $7CBCF. Hang on to the last byte – the $CF – we'll use that as an offset into the page at $DE00. Take the remaining address - $7CB, and turn that into binary: 11111 001011. The low 6 bits should be placed in $DFFE – 001011 is $0B, and the high bits should be placed in $DFFF – 11111 is $1F. The page is now accessible at $DExx - $DECF holds the contents of location 510927.

Let's write a little assembly program to do the register setup for us:

```
    *=$c000
pagelo    .byte 0
pagehi .byte 0
temp   .byte 0

    lda pagelo
    sta temp
    and #%00111111
    sta $dffe

    lda pagehi
    asl temp
    rol
    asl temp
    rol
    sta $dfff
    rts
```

Just store the page number that you want to access in locations $C000/ $C001 (49152/49153) in low/high byte order, and JSR $C003 (SYS 49155). The page is now accessible at $DExx. One other neat feature of the BBGRam I thought of while working on this article: unlike the REU, the BBGRam can actually execute code located within the BBGRam without transferring the memory into the C64 first. Unfortunately, all the code would have to be broken into chunks of 256 bytes – but I'm sure someone could think of a neat use for 2000 or more little routines.

Instead of writing a more generic routine to do many similar tasks (or one large task that needs to be looped many times) one could write a routine that would generate extremely unrolled code for the BBGRam. How about a completely unrolled hires screen clear routine? It wouldn't be faster than the REU, but it could be nearly as fast as a stock machine, without tying up the 24k or so that the code would take.

And thinking about using extra C64 RAM in general (such as SuperRAM), huge tables can speed up a program immensely. For example, an 8-bit times 8-bit multiplication routine, no matter how efficient, will take more cycles than looking up the result in a pre-calculated table. The table would have 65,536 two-byte values, and take up 128k.
Let me know if you come up with some interesting ideas: macbeth@tbaytel.net.

Letters To The Editor

GEOS Help?

Dear Jeff,

On page 8-9 of Loadstar Letter 60 William Mann mentioned lockups in GEOS when moving drive C into drive B position. I have had similar problems that were cured by turning Jiffy-DOS off before booting. Of course, if he does not have JiffyDOS, this is no help.

Dick Estel

Jeff: Hmmm. I've got to see if that works for me. Of course the only catch is that I'm not really that interested in using my Computer without JiffyDOS!

Sensei David O.E. Mohr, Lord Ronin, Responds

Hi Jeff. Thank you for the printing of the picture and the reader of the month section. Comments & feedback on issue #60: Sadly there is no Toys R Us shop in 100 miles of Astoria. I'm hoping that one of my users will be popping by the closest one and can score up a vid cam for me.

Jeff: It's Great to have a reader of the month. I wish more people would send in pictures. By the way, I would also like to see who has the messiest computer desk or computer room. As for the video camera, I'm always glad to get feedback like that.

Ronin: On the other hand. I'm glad that we don't have a Wal-Mart in this area, at least from reading your difficulties with them in digital work.

Jeff: I fixed that with a silly and quite sarcastic letter of permission to myself.

Ronin: If I keep hearing more about Mr. Mouse. I'll just have to score one from CMD and give it a go. I got burned real fast on the 1351. Many problems but it also seems to cause a character drop out when I'm using GEOS. Almost entire words at times, were lost.

Jeff: The way that a mouse ties in to the keyboard can cause characters to be typed. I never heard of this causing a problem in GEOS though. Perhaps you have a version less than 2.0. I'm showing my ignorance here. Did the earlier versions use a proportional mouse in port one?

Ronin: But as to what a "$9000" actually refers too, I don't know. Over my head at this point.

Jeff: $9000 is hexadecimal notation, a numbering system that programmers like to use. See the article after the letters section.

Ronin: OPEN RESPONSE: to the usenet clash article on page #6. My oldest LOADSTAR is #3. Now that I have absorbed the groups LOADSTAR into my library. I've scanned 20 issues of pre-Fender. 30 + issues of post Fender. The programs have steadily gotten better over the years — quality of graphics, speed of the programs, even on non-JiffyDOS drives. Okay, I like some of the effects of graphics and sounds on the older issues when you scrolled too far in a message.

Jeff: I guess if you've never seen the screen bounce like that, it's pretty impressive.

Ronin: Complaints for Loadstar? Yeah I got a few, but I also saw a $69.95 price for a year's subscription to one double-sided disk in 1541 format. We

pay that for four disk sides today. True I think that there should be more for the beginners in Commodore computer understanding. There are words and jargon terms being passed around that make great sense to those in the field for years.

Jeff: I was going to do an article on Assembly language until I printed this part of your letter. Grrr!

Ronin: LOADSTAR needs to jump off its rectum and actively seek out subscribers. NOT those that once subscribed. But new deckers. I run a little hole in the wall 12'x20' shop. I sell back issue comic books, role playing games and support products, and of course the sacred C=64/128 and support items. O.K. we also do a little used Amiga stuff. We are also the meeting spot and GHQ for the users group. I can't count the number of times that I hear *"they still make stuff for the commodore? I loved my 64. Wish I had it again."*

Jeff: We're hoping to get some of these people back through the emulator market.

Ronin: In fact while I was waiting at the hospital for another of those tests — the ones to tell me how much longer I'll suck 02 — I was looking over my Systems book for the 64c. I was asked if I could do windoze. Of course I went into the militant C= user mode. The Doc there told me after that bit that he loved his 128. Did a lot of work with it, even some of his medical studies. Currently it is stuck in the closet, even though it is dead, needing some sort of repair. He was blown away that there were the CMD products, and that LOADSTAR was still around.

Jeff: I get Email like that every week from people who somehow stumble across our website, http://www.loadstar.com

Ronin: I had a couple come in that was really impressed when they saw that rose on LOADSTAR in FLI. Couldn't believe the speed a 64 can go on the Internet till I showed them the copy sheet from CMD. On and on in this vein.

There are thousands of 64s still around. Some hidden away in the attics. But WHY? The users don't know that there is anything for them. WHY IS THAT? Because they look in computer mags. What do they see? That evil drek platform. So if that is all they see, then of course they don't see any C= products. Would it cost that much to have a business size card advert, or even a simple classified advert, pushing Loadstar so they could attract the next generation of Commodore users?

Jeff: Probably a few thousand dollars more than we could afford. Some of these magazines have a circulation of way over a couple million.

Ronin: Just one of the great advantages of the C= machine is that we wanna DO something with it while the other machine does it TO YOU. Just look in the user's guide. You make a program right off. Can you do that with the other platform? People are looking at me strangely. When I show them that you can "program" right on the Commodore PC. While they don't even bother to think about making their stuff. Just canned synthetic stuff made by someone else. CMD - LOADSTAR and other C= supportive companies need to throw a few nuyen into the mainstream advertising market otherwise it is just like selling to yourself. There -- end of spiel.

OH, THE IRONY!

Hi Jeff,

I just finished enjoying another issue of your jam-packed newsletter. Your comments about the problems you were having in Wal-Mart brought to mind a couple of articles I recently read about H.R.2281 that our wonderboy lawyer-leaders in Congress are pushing through. Your problems at Wal-Mart are just the beginning. If you want to read the articles they are at:

www.eagleforum.org/column/1998/june98/98-06-24.html
www.eagleforum.org/column/1998/july98/98-07-29.html

Ron C. Hackley

Jeff: Too bad I can't run those well-written articles about copyright by Phyllis Schafly. Why? Because they're *copyrighted*! But Wal-Mart's copying policy has nothing to do with Congress. It has to do with a sophomoric interpretation of a simple law. The notion that copy**right** laws would take away the **right** of the creator of a work to **copy** his own work is silly.

The notion that a digital copy makes the copy somehow worse, requiring tougher laws is also silly. Since the advent of the disk copy program, people have been able to make perfect digital copies of software for years. With BAM copies, we could even make quicker digital copies, ignoring the stuff we don't need. The sky has not fallen.

It's funny, I am somewhat in the music business these days. One of my coworkers has loaned me CDs for years. I have made technically illegal tapes of his CDs and fallen in love with some and then purchased the ones I wanted. The ones I didn't love ended up being recorded over with my own music. Now that I can duplicate CDs, my coworker will no longer loan me CDs. I find this odd since taping someone else's CD is also considered "mechanical duplication" and just as illegal. In any case, I don't want a duped CD. I want an original, with liners, pressed in aluminum with an indefinite life span instead of the mere hundred years (maybe) that recordable CDs have when stored properly. Because who stores CDs properly?

What Do We Mean When We Say $9000?

By Jeff Jones. Hex is short for HEXADECIMAL, an alternative to the decimal numbering system. Hexadecimal is based on the number 16 instead of 10. So 10 in hexadecimal is actually 16. Decimal numbers are often prefixed with a + sign and hex numbers are prefixed with a $ sign. They are prefixed because some hex numbers can look like decimal numbers when they have entirely different values. For instance a hex value of 80 is actually 128 in decimal, hence the $80 to denote without doubt that we're talking hex. Here's a short chart that counts from 1 to 20 in both number systems:

DECIMAL	HEXADECIMAL
+0	$0
+1	$1
+2	$2

+3	$3
+4	$4
+5	$5
+6	$6
+7	$7
+8	$8
+9	$9
+10	$A
+11	$B
+12	$C
+13	$D
+14	$E
+15	$F
+16	$10
+17	$11
+18	$12
+19	$13
+20	$14

Some people like to prefix hexadecimal numbers with leading zeroes. Instead of $F they may type $0F or even $000F. No difference. This isn't law. It's the same as typing the decimal 15 as 015 or 00015. But why even use hexadecimal? Wasn't decimal good enough? Well, we probably have a decimal numbering system because we have ten fingers and still count on our fingers quite often. Notice how terribly easy it is to count by 10's and by 5's.

Well, your computer is an 8-bit computer. Thinking of each bit as a finger, you can see how a number system based on 8 or 16 would "magically" fall into place, allowing programmers to easily comprehend or remember great spans of RAM. OCTAL-based numbering systems have sprung up, but hex seems dominant.

It's easier to remember that BASIC normally starts and ends at $0800 and $A000 (pronounced "A- thousand") instead of 2048 and 40960. It's also easy to remember when you're writing machine language that I/0 starts at $D000 and KERNAL starts at $E000 (+57344). I honestly don't know offhand where the KERNAL starts in decimal. Let's look at some key C-64 memory locations and the pattern that develops in hex vs. decimal. Which looks harder to remember?

DESCRIPTION	HEX	DECIMAL
BASIC	$0800	+2048
Screen memory	$0400	+1024
Char ROM image	$1000	+4096
Script image	$1800	+6144
Bank 1	$4000	+16384
Bank 2	$8000	+32768
BASIC ROM	$A000	+40960
4K free RAM	$C000	+49152
VIC Chip	$D000	+53248
Border color	$D020	+53280
Background	$D021	+53281
SID Chip	$D400	+54272
KERNAL ROM	$E000	+57344

Note how the hex table is nice and orderly (with the possible exception of the background and border colors). The decimal side is a mishmash of numbers.

Also note that the computer is made up of +256 pages, each page made up of +256 bytes. +256 in hex happens to be $100. So your C-64 has $100 nice clean hex pages of $100 bytes each.

Another boon to using hex which is VERY useful to me is the fact that when you look at a number like $C000 or even $635F, you're looking at the high/low bytes.

Take $C000 (+49152) and break it in half and you get $C0 $00. The Low byte is zero. $C0 = +192. So the low/high bytes are 0/192. Indeed, 192*256+0=49152. How about a number like $635F (+25439)?

Break it in half and we get:

$63 $5F
(+99) (+95)

So the low/high bytes are 95 and 99. 99*256+95=25439 ($635F)

In case you're wondering how I'm able to convert these numbers back and forth between hex and decimal, I use a program. Sometimes I use SUPER AIDE but mostly I use SUPER SNAPSHOT. You're not expected to actually calculate hex math any more than you're expected to know the "secret" code spoken between your TV remote and the set. Conversion is for programs such as monitors and assemblers.

Converting hex to decimal is rather tedious but here's one way:

Look at the decimal number +8128. You have a good grasp of how much 8 thousand is, but what is it really?

10^3	10^2	10^1	10^0
1000	100	10	1
8	**1**	**2**	**8**

8*1000 + 1*100 + 2*10 + 8*1 =8128

That's what 8128 really is: A code for a series of numbers added together. This code is so natural to us that we don't even realize that we decipher it. Now, since hex is based on 16, we simply use 16 as a base (mantissa). Let's convert $642C to decimal using the same method:

16^3	16^2	16^1	16^0
4096	256	16	1
6	**4**	**2**	**C**

6*4096+4*256+ 2*16 + 12*1 = 25644

Not simple? It's just as simple as decimal. It's just that you're not used to it. Again, this is a job for a program. Assemblers and monitors do this for you. You should never have to actually know the values of these numbers. What's more important is that you're able to mentally partition a computer's available memory through hex. That's what it was made for.

Keep Your Variables To A Minimum

By Jeff Jones. When I look at the source code of my older programs, I see a mesh of spaghettiish code laden with needless variables. In the second line of the source to the compiled MAIL MASTER, there is the declaration R=8. I have no idea why I declared that. I think the 8 was used somewhere in the computations for printing envelopes. I listed the first 20 lines of the program and the following variables declared:

```
 ru=1
bs=chr$(20)
nx=15
ra%=6
x$=chr$(13)
b1$="1"
tm=2000
xu$=x$+" "
c=0
pn=4
sa=7
ms$="Press Ctrl-H for help"
h$(6,0)="                    "
```

This was found only in the first screen full of code. The source was 90 blocks in length and contained scores more gratuitous variables. I might add that EXCEPT for PN and SA, which allow the program to set variable secondary addresses and printer device numbers, every single one of these declarations were unnecessary. Actually, PN and SA may be doubly moot because I don't recall giving the user the choice of selecting secondary address or printer

number. There are fundamental mistakes that beginners, and some pros make when coding:

1. Using variables to hold data that never varies or will only be used once.
2. Using variables to flag conditions that don't need a flag at all.
3. Using variables to hold data that is never used.
4. (The worst) Using a different variable for every little subroutine in the program.
5. (A tie for the worst) declaring as a variable text data that you're merely going to print, UNCHANGED, on the next line.

You might ask, "Why limit the use of variables? They're free, aren't they?"

Yes and no. If your program is only 10-40 blocks long, you indeed CAN be carefree and wasteful. But when your program is huge and feature-laden, the program competes with your variables for space. You probably can't spare much of your program, but you can wipe out 90% of your variables.

Variables take up space...not just the data that they store, but the pointers that describe and point to them. Numeric variables take up to seven bytes, even when they are declared to zero. Once declared, a string variable's name and pointers continue to take up space, even if you declare it null. Of course you free up absolutely NO room by nullifying a string if it's declared within the program. That program line where you state:

bl$="   "

is STILL in your program, taking up room. Also, BL$ still exists as a pointer in memory the same way a voided check still exists in your ledger.

I once worked on a program that had so many variables in it that I took a lunch break while X-REF printed me an incredibly long list of them. In almost every case the variables were needless. There had to be at least three hundred variables in the list, not including the individual elements in declared arrays. This program only had about 5 kilobytes free, and probably compromised function because its ceiling had been reached.

The program wouldn't have run out of memory before it finished if that program used 10-20 variables instead of 300. Those 300 variables added up to at least 3000 needlessly used bytes -- more when you consider that scores of these variables were long, long strings.

MANY TIMES YOU DON'T NEED VARIABLES

Example 1

You want to clear a line on the screen. Why use a string defined as 40 spaces? You could just as well print the 40 spaces or use the KERNAL routine at 59903 where you poke 781 with the row number and the sys59903 to clear that row.

Need to plot the cursor anywhere on the screen? Forget those strings defined as HOME followed by a bunch of CRSR downs and CRSR lefts.

```
poke211,x:poke214,y:sys58732
```

This will plot the cursor according to X and Y. Again, unless this is a subroutine to be called by many routines, why use a variable even when I used a variable in the example?

Example 2

I remember a method I used to print messages to the screen. I would declare ms$ as a message and gosub a message routine which would clear any old message from the message area at bottom of the screen, set the color and reverse mode if applicable, center the new message and then return.

Creating message subroutine is a noble effort. Doing so gives your program structure. Most of the needless variables I see in programs are unique messages. Adding a universal message routine that only works with one variable is probably the single most space saving, and variable saving, act you can do.

Let's complicate things a little: I used to use another variable, W, to give the subroutine personality. Make W non-zero and the message routine would print the message and then wait for a key before allowing the program to continue. Could I have done better?

Sure! I could have accomplished the same thing with NO DEDICATED VARIABLES. Here's how:

First, why use ms$ for messages? After all, this routine isn't for only one message. Why use a dedicated variable when the message can be thrown away? So why not use a throwaway variable that's already in use -- the venerable A$? A$ is probably being used to get most keypresses in your program, right. RIGHT!?!? I mean SURELY you're not using A$ to get this key and Q$ to get that key and K$ to wait for this key and J $ ad absurdum -- because such a thing would be ABSURD! CERTAINLY **YOU'RE NOT DOING THAT!?!**

Right?

Anyway, by using the standard throwaway variable, a$, we're down to one dedicated variable in the subroutine. Now we must get rid of W. No problem. W isn't needed at all. In fact, no flag of any sort is needed. Instead of making my message routine "IFfy" by having it wait on a keypress IF W<>0, I could have the message routine always act normally and return without waiting for a keypress. I could simply use a separate subroutine that calls the message routine and then waits for a key.

So if my message routine is at line 2000, I can have another routine at line 2100 that looks something like this:

```
2100 gosub2000:poke198,0
2110 geta$:ifa$=""then2110
2120 return
```

Now if I want to print a message with my message routine, I can put the message in a$ and gosub 2000. If I want to print a prompt as a message and wait for a key, I can put the message in a$, gosub 2100 and then check a$ for the returned keypress.

I could have done that on one line by eliminating the IF command with a WAIT:

```
2100 gosub2100:poke198,1:
wait198,15: geta$:return
```

Program Speed: Too many variables can slow your program. If bw is the 150th variable you created, and nn is the first, it will take roughly 150 times longer for your program to look up bw. Instead of .001 seconds, it might take .15 seconds to access a variable. Still fast? Maybe, but if that variable is in an intense loop, called hundreds of times .15 becomes 15 extra seconds instead of .1 seconds of lookup time. If certain variables will be in routines that require speed, declare those variables first.

WHEN WOULD YOU BREAK THE ABOVE RULES?

Nothing is written in stone -- especially any promise that a new tax is merely temporary. Let's look at exceptions to our new rules.

1. DON'T use variables to hold data that never varies or will only be used once.

Possible exception: placing long numbers into variables will cause repetitious subroutines to go faster. Also sometimes you simply need a constant. 53248 is an example. It is generally agreed that it's okay to poke v+21 to turn sprites off or on rather than poke53248+21. Of course poke 53269 is even better. In a loop it's fastest in a variable. The shorter the variable name, the faster the loop will execute.

Another exception: some formulae are so complex and lengthy that you can't fit them on one line. You may need to perform the computation in stages, using variables.

2. DON'T use variables to flag conditions that don't need a flag at all. To quote the shoemaker, "Just do it."

Possible exception: You simply might need to know later that a certain condition has been met.

3. DON'T declare variables to hold data that is never used by the program.

Possible exception: Possible debugging formula. Or God may tell you to do it for some reason we humans cannot fathom.

4. DON'T make up a new variable for every little subroutine in the program.

Possible exception: The devil told you to do it.

5. DON'T declare as a variable any text data that you're merely going to print, UNCHANGED, later. Just print it.

Possible exception: You may want to print to the screen as well as the printer.

6. DO remember that once declared, any variable continues to take up SOME space even after you nullify it or zero it out.

Possible exception: None.

7. DO try to use the same variable names for multiple purposes as much as possible.

Possible exception: You may want to remember what you have done in the past. That is, if your program goes into modes where it should do "this" if it has previously done "that." Writing true subroutines that can be used by OTHER subroutines may seem complicated, but it makes coding easier. The worst quagmire you can get into is the BIG routine that does "this" as well as "that" depending on personality-controlling variables.

Arrgh! I Got Burned By Little Red Reader!

By Jeff Jones. Well, not really burned. I created some ASCII text files for port over to my PC computer using my Loadstar Issue Text Extractor. It produced three sequential files. I copied them over to an MS-DOS disk and drove halfway across town only to find three corrupt files.

Turns out that Little Red Reader assumes that SEQ files are PetASCII files and automatically sets them as ASC transfers. That's all right except that my files were already translated. So when Little Red Reader tried to convert what was already ASCII into ASCII, Only the uppercase, numbers and punctuation survived.

I was afraid that it was a scrambled RAMLink until I figured it out.

BASIC Stuff

By Jeff Jones. I'm often accused of being too technical. Here's the first installment of my Beginner's series. Anyone who's thinking about programming should read this — and then order the Compleat Programmer from Loadstar!

When I program, I feel as if I were "coaxing the computer to do exactly what I want by means of instructions." In actuality anyone who is thoughtful, can plan ahead, and isn't in the habit of abandoning projects, can write flawless software and eventually be published on LOADSTAR.

"Do this then do that, then add this to that then print it" is the general scheme to programming. We instruct computers by means of a computer language. The language can be BASIC, MACHINE LANGUAGE, PASCAL, C, etc. But we're going to work with BASIC in these early chapters.

BASIC was meant to be an easy "sample" language, used to teach logic in colleges. We always capitalize BASIC because it's an acronym for BEGINNER'S ALL-PURPOSE SYMBOLIC INSTRUCTION CODE. Needless to say, BASIC is used for many more things than logic courses. It was found that BASIC could be quite powerful. It had been enhanced time and again. The BASIC of today isn't the BASIC that's taught in colleges, even today's colleges in some cases. Professors might mention that "BASIC can't" when today's BASIC can -- especially when you compile it! More on compiling in later chapters.

IMMEDIATE MODE

Unless a program is RUNning, your cursor should be flashing. This usually means you're in the IMMEDIATE MODE. IMMEDIATE MODE programs only exists on the screen, not in BASIC's storage area.

You can make a makeshift spreadsheet in the IMMEDIATE MODE. Let's say that your paycheck is $500 and you want to know how much money you'll have left over after all your bills. Though there are many BASIC commands that work in the IMMEDIATE MODE, we'll only concern ourselves with one for now.

How many commands does it take to handle a paycheck and a few bills? How about one? The ever-popular PRINT command will handle your bills in a breeze! Just turn on your computer and type:

```
PRINT 500
```

$500 is your paycheck. Press RETURN and you should see:

```
500
READY.
```

What just happened? You asked BASIC to PRINT 500 -- and it did. Now type this:

```
PRINT 500-100
```

Press RETURN and you'll see:

```
400
READY.
```

You just asked BASIC to print 500 minus 100. BASIC, your loyal servant, did a little math, figured out the solution to 500 minus 100, then printed it on the screen.

```
?SYNTAX ERROR
```

Did you misspell PRINT as PRONT? Whenever you see ?SYNTAX ERROR, it means the computer came across a command that it couldn't understand. SYNTAX means about the same as GRAMMAR. So when the computer says this, it really means "Geeze! Can't you speak good BASIC?

How am I supposed to understand that?" Think of the computer as a snooty 6th grade English teacher that can't abide ANY slang. The computer has a limited vocabulary of about 60 words. It can't guess what you really meant.

Wanna figure out how much you'll have left after a few bills?

```
PRINT 500-112-34.45-12.98-15.99-25
```

Press RETURN and you'll see:

```
299.58
READY.
```

Because the C-64 has the world's BEST (I mean this!) screen editor, you can use the cursor keys to move back up and edit this line to add or take away from (juggle) your bills.

Let's get cosmetic:

```
PRINT "$"500-112-34.45-12.98-15.99-
25
```

Press RETURN and you'll see:

```
$ 299.58
READY.
```

Whatever you place within quotes will be printed as you defined it within the quotes. Since we placed a dollar sign in quotes, it printed the dollar sign, then printed the solution to your budget problem.

You'll find that PRINT is very powerful and very commonly used. It's rather complicated to place or remove anything on the screen without resorting to the PRINT command.

VARIABLES

You can figure your bills without using the PRINT command:

```
X = 500-112-34.45-12.98-15.99-25
```

Of course you can't see X yet. You'll have to eventually PRINT it.

Here we introduce variables. X is a variable, meaning that it can represent most any number that you tell it to. The largest number X can represent is 1.70141183 times 10 to the 38th power. Or 1.7 times 10 followed by 38 zeroes. Hopefully, with Clinton's deficit reduction package, no bill will ever be that high.

```
PRINT X  <press RETURN>
299.58
READY.
```

You might have noticed that there is always a space in front of PRINTed numbers. Think of that space as an invisible plus sign. If the number were negative, there would be a visible negative sign.

STRING VARIABLES

String variables hold text instead of numbers. The variables are called "strings". A$ reads A- STRING. BF$ reads as BF-STRING. Let's watch them in use. For instance:

```
A$ = "LOADSTAR"
```

This command will make A$ represent "LOADSTAR. Now try this:

```
PRINT A$  <RETURN> LOADSTAR
READY.
PRINT A$;A$
LOADSTARLOADSTAR
READY.
PRINT A$ " " A$
LOADSTAR LOADSTAR
READY.
```

Note that the computer did exactly what you asked each time. In the second command, we printed A$ twice. Note that we separated them with a semicolon. In this case it wasn't needed but in some cases such as the following, it could be:

```
PRINT A$BA$NA$
```

Here you might want to print A-string, BA-string, then NA-string. If so, then everything's fine and dandy. But you might actually want the computer to print A-string, B- variable, A-string, N-variable, and A-string again.

For this task you'll need semicolons in order to warn the BASIC interpreter of your intentions. A semicolon is a separator, allowing you and the computer to distinguish between variables that are heaped together. Commas can also be used as separators, but they force tabs to the nearest tenth column.

```
PRINT A$B;A$N;A$
```

or

```
PRINT A$;B;A$;N;A$
```

The semicolons are ignored and both accomplish the same goal. But WITHOUT semicolons, you'd get totally different results in this case.

```
A$="LOADSTAR"
READY.
PRINT A$" "A$
LOADSTAR LOADSTAR
READY.
PRINT A$,A$
LOADSTAR  LOADSTAR
READY.
```

PRINT A$" "A$, literally says, "print the A-string variable, then print a space, then print A-string again."

PRINT A$,A$ means print A$, then tab to the nearest tenth column (10, 20, 30, 40) and print A$ again.

Okay, we've talked about the IMMEDIATE MODE. Now on to actual programs. A program is "a series of instructions..." To keep these instructions in order, we label them with LINE NUMBERS. BASIC automatically sorts your lines. there is no need to enter them in order.

```
10 PRINT "HELLO"
20 END
```

So you would enter your lines exactly as shown and press <RETURN> when done. In order to try your programs, type RUN and press <RETURN>

To erase line 10, type 10 <RETURN> and line 10 is gone. The same for any other line. If you want to type over a line, you can either re-type it or type LIST <RETURN> and then move your cursor up to the line and edit the line and press <RETURN> on the line. LIST will list your entire program to the screen. LIST 10 will only list line 10. LIST 50- 70 will only show lines 50-70.

When you're ready to try a different program and want to clear out the program in memory, type NEW and <RETURN>.

Please don't type in caps. We present some text in caps to make it stand out. Generally, the only time you'll type anything in caps is when you're printing text to the screen.

The preceding program consists of two lines, 10 and 20. When you enter a program line, the space between the line number and the first command is inserted for you. Line numbers are automatically kept in order. To erase a line just type the number of the line and press RETURN.

To start a program, type RUN. The program will start running at the lowest line number, line 10. It will print HELLO and then end on line 20.

You can add more lines:

```
10 I = 1
20 PRINT "HELLO" I
30 I = I + 1
40 IF I < 20 THEN GO TO 20
50 END
```

Note the less than symbol. We learned this symbol in 2nd grade. It means the same thing in programming as it did in the 2nd grade. No astrophysics here. If I is less than 20 then the program will go to line 20. If I is greater than 20 then everything after the THEN statement will be skipped and the program will continue on the next line.

RUN this program and it will print HELLO 1 through HELLO 20 and then end. Line 10 sets I to 1. Line 20 prints "hello" and the value of I. Line 30 increments I. Line 40 introduces CONDITIONAL BRANCHING, using the IF - THEN combination.

The way IF THEN works is simple. You set up a condition, in this case when I is less than 20. If this condition is true, BASIC will execute the commands after the THEN command. If not, BASIC falls through to the next line. Though not well illustrated in this example (since the program ends), it's a good idea to make sure that your program properly handles what happens when an IF THEN test is not true.

The typical program flows from one command to the next. This is called SEQUENTIAL FLOW. Sometimes you might want a program to JUMP or GO TO a line out of sequence. This is called BRANCHING. Other times you want to branch only IF a condition exists or doesn't exist. This is called CONDITIONAL BRANCHING. At other times you may want the program to perform a task again and again before moving on. This is called LOOPing. Look for these types of flow as we compute in this lesson. If you look closely, you'll see combinations of these basic types of program flow.

FOR NEXT loops are very useful, and at least one will be found in most any program -- in many languages.

DEFINITION: A method of performing repetitive tasks with a minimum of BASIC code, using FOR, TO and NEXT commands.

```
10 LET I = 1
20 PRINT "TEST" I
30 I=I+1
40 IF I < 10 THEN GOTO 20
50 END
```

Note that I used the LET command in line ten. LET is an absolutely useless command. In olden days it used to herald the declaration of a variable. Now all it does is take up space since I = 1 will work without LET. The equal sign makes the intention obvious to the C-64. You have my official permission to absorb the purpose of LET -- then shelve it forever. So why is the command there when the makers of the C-64 knew it wasn't needed? It's part of "standard BASIC". You couldn't call BASIC BASIC with LET absent. Besides some old generic programs have LET in them and wouldn't RUN on a computer that didn't recognize LET.

The program above will print TEST 10 through TEST 10 on your screen without using a FOR NEXT loop. Let's try it with a FOR NEXT loop and see how the code changes.

```
10 FOR I = 1 TO 10
20 PRINT "TEST" I
30 NEXT
40 END
```

This code accomplishes the same task, though with more finesse. The entire program could have been written on one line. The previous version with the manual loop required at least three lines because of the one-time initialization of the variable I and the IF THEN conditional.

By the way, the END command return you to BASIC. The program would have returned to BASIC (or ENDed) once it ran its course anyway. Sometimes programmers might place a STOP command in their code, which is at times more useful because it tells you which line the program stopped on.

To place more than one command on a line, use colons:

```
10 FOR I=1 TO 10: PRINT"TEST"I:
NEXT
```

Voila! The same program running just a tinch faster.

The variable "I" was used in the FOR NEXT loop. I was used to count in the loop. Everything between the FOR TO and NEXT is repeated until I equals 10.

NOTE! When the loop is finished, I will be 11. The loop won't end until I is out of range. That's at 11, not 10. That little quirk may come back and nip you if you try to use the I variable after the loop and expect it to be 10.

Note that, when using a FOR NEXT loop, you don't have to initialize variables, check for a maximum value or tell the program which line to go back to. There can be many, many lines enclosed within a FOR NEXT loop.

NEXT increments I every time it's called, until I reaches the maximum value declared with the TO command. Once the loop reaches the maximum iteration, the NEXT will allow program flow to pass it, and the program will continue beyond the NEXT command. You can have commands appear after NEXT, on the same line though many like to set NEXT apart in order to clearly mark the boundaries of the loop.

STEP

What if you only want to count even numbers or odd numbers, or by fives? That's where STEP comes in:

```
 10 FOR I = 2 TO 10 STEP 2
20 PRINT I
30 NEXT
```

STEP 2 tells the FOR NEXT loop to count in steps of +2 instead of the default +1. So every time NEXT is encountered, it will add +2 to I, resulting in the following output:

```
 RUN
2
4
6
8
10
 READY.
```

What if you want to count backwards?

```
 10 FOR I = 5 TO 0 STEP -1
20 PRINT I
30 NEXT
```

Now every time NEXT is encountered, -1 will be added to I, resulting in a decrement instead of increment.

COMMODORE BBS LIST

Starting on the next page is a list of all known OPEN Commodore Bulletin Board Systems operated on GENUINE Commodore 64 or 128 computers, sorted by country and telephone number. The list is published at least once per month (preferably twice per month), with no set schedule. The latest copy is always available at ftp://jbrain.com/pub/cbm/faq/cbm-bbs-list.txt (normally within minutes of arriving at the server).

If you would like to receive a copy of this list each time it is published, send a new email message to wanderer@cyberdude.com.

Verified Active Commodore BBS List

Hidden Empire — 201-460-7955

Woodridge, New Jersey BBS Software/Networks....C-Net 128/CommNet
BBS Platform/Hardware....C128/CHD
System Operator Name.....Patrick Domanski (patrick873@aol.com)
System Operator Handle...Polish Warrior
Maximum Modem Speed.....2400 BPS
Open Status Verified.....4 Jun 1998 - Ron Fick

The Valley — 206-840-1031

Location.....Puyallup, Washington
BBS Software/Networks....Image/CommNet
BBS Platform/Hardware....C64
System Operator Name.....Steve Johnson
System Operator Handle...Roller Man
Maximum Modem Speed.....14.4K BPS
Open Status Verified.....4 Jun 1998 - Ron Fick

Pink Panther — 208-587-7636

Mountain Home, Idaho
BBS Software/Networks....Omni 128/Omni EchoNet
BBS Platform/Hardware....C128
System Operator Handle...Mad Max
Maximum Modem Speed.....14.4K BPS
Open Status Verified.....20 Nov 1997 - Ed Paulsen

Silicon Realms — 209-754-1363

San Andreas, California
BBS Software/Networks....Image/CommNet, XNet
BBS Platform/Hardware....C64/CHD
System Operator Name.....Larry Anderson (foxnhare@goldrush.com)
System Operator Handle...Joe Commodore
Maximum Modem Speed.....14.4K BPS
http://www.goldrush.com/~foxnhare/srlinfo.html
Open Status Verified.....4 Jun 1998 - Ron Fick

Never Ending Warez — 248-745-8259

Pontiac, Michigan
BBS Software/Networks....C-Net 128/CommNet
BBS Platform/Hardware....C128/LtK
System Operator Name.....Ron Emery
System Operator Handle...C-Net 1/CS
Maximum Modem Speed.....2400 BPS
Open Status Verified.....4 Jun 1998 - Ron Fick

Omni World 128 — 253-536-9353

Parkland, Washington
BBS Software/Networks....Omni 128/Omni EchoNet
BBS Platform/Hardware....C128
System Operator Name.....Brian Bell
System Operator Handle...Dr. Midi
Maximum Modem Speed.....14.4K BPS
Open Status Verified.....20 Nov 1997 - Ed Paulsen
Omni 128 BBS HQ

The Coffee Shop — 253-565-6306

Fircrest, Washington
BBS Software/Networks....Omni 128/Omni EchoNet
BBS Platform/Hardware....C128
System Operator Name.....Unknown
System Operator Handle...P.I.
Maximum Modem Speed.....14.4K BPS
Open Status Verified.....19 Jul 1998 - Pat Keller, Jr.

Harry's Asylum — 281-471-6503

Laporte, Texas
BBS Software/Networks....Image
BBS Platform/Hardware....C64/LtK
System Operator Name.....Glenn Carman
System Operator Handle...Dr. Harry
Maximum Modem Speed.....14.4K BPS
Open Status Verified.....4 Jun 1998 - Ron Fick
Oldest C= BBS in Houston

Bateave — 303-252-0735

Denver, Colorado
BBS Software/Networks....C-Net 128/CommNet
BBS Platform/Hardware....C128/LtK
System Operator Name.....Ron Fick (rfick@myx.net)
System Operator Handle...Caped Crusader
Maximum Modem Speed.....2400 BPS
Open Status Verified.....4 Jun 1998 - System Operator
The only source for new Lt. Kernal HD components
1 of 2 main network hubs for C-Net 128 CommNet gateway.

Land of Oz — 303-985-3980

Lakewood, Colorado
BBS Software/Networks....C-Net 128/CommNet
BBS Platform/Hardware....C128/LtK
System Operator Name.....Don Koblischke (dkoblis@worldnet.att.net)
System Operator Handle...Gandalf the Gray
Maximum Modem Speed.....2400 BPS
Open Status Verified.....4 Jun 1998 - Ron Fick

Inner Circle — 304-697-0101

Huntington, West Virginia
BBS Software/Networks....Centipede/ComLink, CommNet, Net64
BBS Platform/Hardware....C128/16MB RL, 3.2GB CHD, SC128
System Operator Name.....John Pinson (scebbs@ramlink.net)
System Operator Handle...Iceman/ICE (formerly Top Cop)
Maximum Modem Speed.....38.4K BPS
http://ram.ramlink.net/~scebbs/
Open Status Verified.....16 Jun 1998 - Dick Cunningham
ICE HQ, one of the world's largest Commodore BBS's.

DiamondBack — 305-258-5039

Miami, Florida
BBS Software/Networks....C-Net 128/CommNet
BBS Platform/Hardware....C128/CHD
System Operator Name.....Mike Egleston
System Operator Handle...SMS Mike
Maximum Modem Speed.....2400 BPS
Open Status Verified.....4 Jun 1998 - Ron Fick

Get It Here — 309-764-7084

Moline, Illinois
BBS Software/Networks....C-Net 128/CommNet
BBS Platform/Hardware....C128
System Operator Name.....Chris Hailer (the-mage@worldnet.att.net)
System Operator Handle...The Mage
Maximum Modem Speed.....2400 BPS
Open Status Verified On..4 Jun 1998 - Ron Fick

The Blade Shop — 312-434-0142

Chicago, Illinois
BBS Software/Networks....V128
BBS Platform/Hardware....C128
System Operator Handle...Unknown
Maximum Modem Speed.....2400 BPS
Open Status Verified.....25 Oct 1997 - Kenneth Nealey

Palisades — 314-451-6901

Pacific, Missouri
BBS Software/Networks....Centipede/ComLink
BBS Platform/Hardware....C128
System Operator Name.....Ron Hooper
System Operator Handle...Fatboy
Maximum Modem Speed.....14.4K BPS
Open Status Verified.....1 Feb 1998 - Dick Cunningham

C64 Micro BBS — 317-299-2198

Indianapolis, Indiana
BBS Software/Networks....Written by sysop/no network
BBS Platform/Hardware....C64/CHD
System Operator Name.....Barry Skidmore (bskidmore@iquest.net)
System Operator Handle...Barry Skidmore
Maximum Modem Speed.....2400 BPS
Open Status Verified.....2 Aug 1998 - System Operator
No validation required for access

The Phoenix — 360-373-2715

Bremerton, Washington
BBS Software/Networks....Omni 128
BBS Platform/Hardware....C128
System Operator Handle...Radar
Maximum Modem Speed.....14.4K BPS
Open Status Verified.....30 Oct 1997 - Fungus - F4CG/Carcass
Support for VIC20, C64, C128, Amiga

Future Frontier — 360-384-4704

Ferndale, Washington
BBS Software/Networks....Image/CommNet
BBS Platform/Hardware....C64
System Operator Name.....Bob Sisco (bsisco@az.com)
System Operator Handle...Iron Axe
Maximum Modem Speed.....14.4K BPS
Open Status Verified.....19 Jul 1998 - Pat Keller, Jr.

Castle Royale — 360-647-7120

Bellingham, Washington
BBS Software/Networks....Image/Future Net to CommNet, WorldNet, XNet
BBS Platform/Hardware....C64
System Operator Name. ...Terry Asp (schlangel@aol.com)
System Operator Handle...Schlange
Maximum Modem Speed.....14.4K BPS
Open Status Verified On..4 Jun 1998 - Ron Fick

The Club House — 360-675-7172

Oak Harbor, Washington
BBS Software/Networks....V128
BBS Platform/Hardware....C128
Maximum Modem Speed.....9600 BPS
Open Status Verified.....25 Oct 1997 - Kenneth Nealey

First Contact — 402-393-2985

Omaha, Nebraska
BBS Software/Networks....C-Net 128/CommNet
BBS Platform/Hardware....C128/LtK
System Operator Name.....Ken Ringenberg (valdar2@aol.com)
System Operator Handle...Valdar
Maximum Modem Speed.....2400 BPS
Open Status Verified.....4 Jun 1998 - Ron Fick
Club BBS for GOCUG.

Digital Power — 405-672-8995

Oklahoma City, Oklahoma
BBS Software/Networks....Image
BBS Platform/Hardware....C64
Maximum Modem Speed.....9600 BPS
Open Status Verified.....25 Oct 1997 - Kenneth Nealey

Meg II's Altitude — 405-793-9892

Moore, Oklahoma
BBS Software/Networks....Image
BBS Platform/Hardware....C64/LtK
System Operator Name.....Mark Page
System Operator Handle...Flyboy
Maximum Modem Speed.....14.4K BPS
Open Status Verified.....4 Jun 1998 - Ron Fick

Boots — 414-437-9970

Green Bay, Wisconsin
BBS Software/Networks....V128
BBS Platform/Hardware....C128
Maximum Modem Speed.....2400 BPS
Open Status Verified.....25 Oct 1997 - Kenneth Nealey

Dumbo's Flying Circus — 414-521-2440

Waukesha, Wisconsin
BBS Software/Networks....Image/CommNet
BBS Platform/Hardware....C64
System Operator Name.....Kevin Cook
System Operator Handle...Dumbo
Maximum Modem Speed.....14.4K BPS
Open Status Verified.....4 Jun 1998 - Ron Fick

Ultimate Force — 415-441-1120

San Francisco, California
BBS Software/Networks....Centipede/ComLink, Net64 C128/16MB RL, 4GB CHD
System Operator Name.....Kenneth Nealey (wizard_2@pacbell.net)
System Operator Handle...Mega Force
Maximum Modem Speed.....14.4K BPS
http://home.pacbell.net/wizard_2
Open Status Verified.....16 Jun 1998 - Dick Cunningham
Largest Commodore BBS in Northern California

Home Port — 425-334-8298

Everett, Washington
BBS Software/Networks....V128
BBS Platform/Hardware....C128
System Operator Handle...Sailor
Maximum Modem Speed.....14.4K BPS
Open Status Verified.....25 Oct 1997 - Fungus - F4CG/Carcass

Lucky — 502-933-5397

Louisville, Kentucky
BBS Software/Networks....C-Net 128/CommNet
BBS Platform/Hardware....C128/LtK
System Operator Name.....Dave Snyder
System Operator Handle...Tenpenny
Maximum Modem Speed.....2400 BPS
Open Status Verified.....4 Jun 1998 - Ron Fick
Club BBS for LUCKY

Jim's Room — 503-254-6011

Portland, Oregon
BBS Software/Networks....Color SML
BBS Platform/Hardware....C64
Maximum Modem Speed.....2400 BPS
Open Status Verified.....25 Oct 1997 - Kenneth Nealey

Radio Tower — 503-722-8009

Milwaukee, Oregon
BBS Software/Networks....Centipede/ComLink
BBS Platform/Hardware....C128/JD,REU,500MB CHD
System Operator Name.....David Erickson (david.erickson@pods.pacifier.com)
System Operator Handle...John Denver
Maximum Modem Speed.....33.6K BPS
Open Status Verified.....18 Jul 1998 - System Operator
Free membership, all platform support in downloads, Christian theme BBS

The 128 P.C. — 512-940-0023

Kyle, Texas
BBS Software/Networks....C-Net 128
BBS Platform/Hardware....C128/1.6K
System Operator Name.....Tom Perariteau (tomp@gte.net) System Operator Handle...Tom Perariteau
Maximum Modem Speed.....14.4K BPS
http://home1.gte.net/tomp/
Open Status Verified....4 Jun 1998 - Ron Fick
The only C= BBS left in the Austin metro area. Also the only known C= BBS using a cellular phone for access.

Beaky — 518-783-1631

Cohoes, New York
BBS Software/Networks....Color SML
BBS Platform/Hardware....C64
Maximum Modem Speed.....2400 BPS
Open Status Verified.....25 Oct 1997 - Kenneth Nealey

Twilight Zone — 602-827-2706

Tempe, Arizona
BBS Name BBS Software/Networks ...V128/ComLink, Net64
BBS Platform/Hardware C128/SwiftLink, 16MB RL, 1GB CHD
System Operator Name Tim Allen (dynamite@mindspring.com)
System Operator Handle...Dynamite
Maximum Modem Speed.....28.8K BPS (Hayes Optima 28.8)
WWW Presence http://dynamic.home.mindspring.com
Open Status Verified.....16 Jun 1998 - System Operator
Comments Color 64 V7 Distribution BBS

Bass Planet — 609-587-4495

Mercerville, New Jersey
BBS Software/Networks....C*Base 64
BBS Platform/Hardware....C64
Maximum Modem Speed.....2400 BPS
Open Status Verified.....25 Oct 1997 - Kenneth Nealey

The Mailbox — 610-834-9694

Conshohocken, Pennsylvania
BBS Software/Networks....Image/CommNet
BBS Platform/Hardware....C128-64/16MB RL, 85MB CHD, SC64
System Operator Name.....Nelson Schrock (keymaster14@juno.com)
System Operator Handle...Keymaster
Maximum Modem Speed.....14.4K BPS (USR 33.6)
http://members.aol.com/KeyNelse/mailbox.htm
Open Status Verified.....15 Aug 1998 - System Operator

The Last Stand — 612-533-5267

Minneapolis, Minnesota
BBS Software/Networks...Centipede/ComLink
BBS Platform/Hardware....C128
System Operator Name.....Mike Martin (mmartin@pclink.com)
System Operator Handle...Voyager
Maximum Modem Speed.....14.4K BPS
Open Status Verified.....16 Jun 1998 - Dick Cunningham

Gremline — 616-795-8706

Middleville, Michigan
BBS Software/Networks.....Image 1.2a/CommNet
BBS Platform/Hardware....C128/SL,CHD 170MB
System Operator Name.....Mark Newman
System Operator Handle...Gremline
Maximum Modem Speed.....14.4K BPS (USR Modem)
Open Status Verified.....11 Sep 1998 - Jim Caldwell

Cereal City — 616-962-1390

Battle Creek, Michigan
BBS Software/Networks....Omni 128
BBS Platform/Hardware....C128
Maximum Modem Speed.....2400 BPS
WWW Presence.....None
Open Status Verified.....25 Oct 1997 - Kenneth Nealey

Golden Reef — 619-390-0351

Lakeside, California
BBS Software/Networks....C-Net 128/CommNet
BBS Platform/Hardware....C128/LtK
System Operator Name.....Betty McKnight
System Operator Handle...Fisheye
Maximum Modem Speed.....2400 BPS
Open Status Verified.....4 Jun 1998 - Ron Fick

Dream Factory — 619-390-7483

Lakeside, California
BBS Software/Networks....C-Net 128/CommNet
BBS Platform/Hardware....C128/CHD
System Operator Name.....Gordon Wright (chameleon@juno.com)
System Operator Handle...Chameleon/CSD
Maximum Modem Speed.....2400 BPS
Open Status Verified.....4 Jun 1998 - Ron Fick

Commodore Corner — 619-523-1812

San Diego, California
BBS Software/Networks....New Image v1.2
BBS Platform/Hardware....C64
System Operator Name.....Brent Hendricks
System Operator Handle...Rug Rat
Maximum Modem Speed.....2400 BPS
Open Status Verified.....4 Jun 1998 - Ron Fick
Only known C= BBS operated from on a boat!

The Abyss — 619-874-6921

San Diego, California
BBS Software/Networks....Centipede/ComLink, CommNet
BBS Platform/Hardware....C128/2GB CHD
System Operator Handle...Eddie
Maximum Modem Speed.....28.8K BPS
Open Status Verified.....4 Jun 1998 - Ron Fick
They're Back!

Den of Iniquity — 702-365-9036

Las Vegas, Nevada
BBS Software/Networks....Color 64
BBS Platform/Hardware....C128-64/2 1571's
System Operator Handle...Mr. Fixit
Maximum Modem Speed.....2400 BPS
Open Status Verified.....2 Sep 1998 - Hernan Vergara

First Blood — 702-399-2415

Las Vegas, Nevada
BBS Software/Networks....Color 64
BBS Platform/Hardware....C128-64/5GB Storage
System Operator Handle...Big Bob
Maximum Modem Speed.....2400 BPS
Open Status Verified.....2 Sep 1998 - Hernan Vergara
One of the first BBSs in the U.S.

Sunlight — 702-673-2927

Sun Valley, Nevada
BBS Software/Networks....V128/Net64
BBS Platform/Hardware....C128
System Operator Name.....William Creveling IV
System Operator Handle...Shadow Blue
Maximum Modem Speed.....14.4K BPS
Open Status Verified.....8 Feb 1998 - Hernan Vergara

Seek and Destroy — 703-669-1244

Bristol, Virginia
BBS Software/Networks....V128
BBS Platform/Hardware....C128
Maximum Modem Speed.....2400 BPS
Open Status Verified.....25 Oct 1997 - Kenneth Nealey

North Pole — 708-986-1295

Darien, Illinois
BBS Software/Networks....V128
BBS Platform/Hardware....C128
Maximum Modem Speed.....2400 BPS
Open Status Verified.....25 Oct 1997 - Kenneth Nealey

Nature Reserve — 714-828-7296 (Line 1)

Cypress, California
714-952-2696 (Line 2 which is ComLink Hub from 11 PM - 6 AM PST)
BBS Software/Networks ...Centipede/ComLink
BBS Platform/Hardware ...C128/multiplexed LtK
System Operator Name.....Adam Fanello (adam@acm.org)
System Operator Handle...Ant
Maximum Modem Speed.....14.4K BPS (Both Ports)
http://www.bugsoftware.com/
Open Status Verified.....16 Jun 1998 - Dick Cunningham
Centipede & V128 HQ

The Huntington Connection — 714-848-4692

Huntington Beach, California
BBS Software/Networks....Image
BBS Platform/Hardware....C64
Maximum Modem Speed.....2400 BPS
Open Status Verified.....25 Oct 1997 - Kenneth Nealey

Conquer the City — 718-680-8038

Brooklyn, New York
BBS Software/Networks....Omni 128
BBS Platform/Hardware....C128
Maximum Modem Speed.....9600 BPS
Open Status Verified.....25 Oct 1997 - Kenneth Nealey

Portsmouth Commodore Users Group — 804-393-2949

Portsmouth, Virginia
BBS Software/Networks....Color 64
BBS Platform/Hardware....C64
Maximum Modem Speed.....2400 BPS
Open Status Verified.....25 Oct 1997 - Kenneth Nealey

Ski Resort — 804-393-4964

Portsmouth, Virginia
BBS Software/Networks....Color 64
BBS Platform/Hardware....C64
Maximum Modem Speed.....2400 BPS
Open Status Verified.....25 Oct 1997 - Kenneth Nealey

Channel 19 — 804-422-4671

Virginia Beach, Virginia
BBS Software/Networks....V128
BBS Platform/Hardware....C128
Maximum Modem Speed.....14.4K BPS
Open Status Verified.....25 Oct 1997 - Kenneth Nealey

Network 23 — 804-431-2854

Virginia Beach, Virginia
BBS Software/Networks....Color SML
BBS Platform/Hardware....C64
Maximum Modem Speed.....14.4K BPS
Open Status Verified.....25 Oct 1997 - Kenneth Nealey

Civic 64/128 BBS — 805-382-1125

Oxnard, California
BBS Software/Networks....Omni 128/Omni EchoNet
BBS Platform/Hardware....C128/1.28GB CHD
System Operator Name.....Ben Holmes (bnholmes@rain.org)
System Operator Handle...Ben Holmes
Maximum Modem Speed.....33.6K BPS
Open Status Verified.....15 Aug 1998 - System Operator
Many Utilities

Starfleet Command — 810-258-9864

Birmingham, Michigan
BBS Software/Networks....C-Net 64 DS2/CommNet
BBS Platform/Hardware....C64
System Operator Name.....Kerry Borgne (deaconone@teleweb.net)
System Operator Handle...Deacon One
Maximum Modem Speed.....2400 BPS
Open Status Verified.....4 Jun 1998 - Ron Fick

Magic's Realm — 813-986-6786

BBS Software/Networks....C-Net 64 DS2/CommNet
BBS Platform/Hardware....C64
System Operator Name.....Doug Hollingsworth (prophet3@gte.net)
System Operator Handle...Demon
Maximum Modem Speed.....2400 BPS
Networks.....CommNet
Open Status Verified.....4 Jan 1998 - Ron Fick

The RoadHouse! — 818-265-9888

Glendale, California
BBS Software/Networks...Centipede/ComLink, Net64
BBS Platform/Hardware....C128/JD, 170MB CHD
System Operator Name.....Sean Kilian (cpnknight@earthlink.net)
System Operator Handle...Captain Knight
Maximum Modem Speed.....57.6K BPS
Open Status Verified.....1 Feb 1998 - Dick Cunningham
Messages, Games

Live Zone — 901-584-2588

Camden, Tennessee
BBS Software/Networks....C-Net 128/CommNet
BBS Platform/Hardware....C128
System Operator Name.....Lloyd Nelms
System Operator Handle...Night Person
Maximum Modem Speed.....2400 BPS
Open Status Verified.....4 Jun 1998 - Ron Fick

STAR-Link — 914-621-4135

Mahopac, New York
BBS Software/Networks C-Net 128/CommNet BBS Platform/Hardware C128
System Operator Name Eric Pearson (C-Net128@aol.com)
System Operator Handle Darth
Maximum Modem Speed 14.4K BPS
Networks CommNet
http://ourworld.compuserve.com/homepages/cnet128/cnethome.htm
http://members.aol.com/cbmroot65/cnet1.html (frames)
Open Status Verified 4 Jun 1998 - Ron Fick
Owner of C-Net 128 BBS program

Star Gate — 919-603-5916

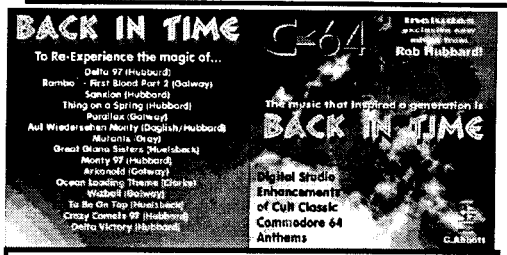
Oxford, North Carolina
BBS Software/Networks....Supra 128
BBS Platform Hardware....C128
System Operator Name.....Donald Coombs (stargate@glorvroad.net)
System Operator Handle...Swamp Rat
Maximum Modem Speed.....9600 BPS
Open Status Verified.....5 Sep 1998 - System Operator
Up since 1988 (excluding 1 year downtime)

Starfleet Command BBS — 919-732-9309

Telephone Number
Durham, North Carolina
BBS Software/Networks Centipede/ComLink, Net64
BBS Platform/Hardware C128/SwiftLink, 10B CHD
System Operator Name Matthew Price (admyank@worldnet.att.net)
System Operator Handle Admiral Yank
Maximum Modem Speed 19.2K BPS
http: home.att.net/~admyank
Open Status Verified 13 Jun 1998 - System Operator
North Carolina's only Star Trek oriented Commodore BBS supporting multiple platforms and amateur radio operators

The Dungeon — 931-648-0577

Indian Mound,TN
BBS Software Networks....C*Base v3.2 C*Base Specific Network
BBS Platform/Hardware....C128-64 4MB RL parallel to 120MB CHD
System Operator Name.....Currey Handle...Scorpio
Maximum Modem Speed.....19.2K BPS (14.4K Modem)
http: www.public.usit.net/scorpio
Open Status Verified.....16 Nov 1997 - System Operator
Files section, games, message bases



Available from LOADSTAR!

Chris Abbot's goal was to professionally reproduce well-loved Commodore demo and game tunes. He pulls this off quite well, using state-of-the-art MIDI equipment. These CDs were not manufactured on a PC's CD recorder. They were professionally pressed, fully packaged and contain a nice little booklet with explanations for each song along with a Rob Hubbard interview. You should get this CD, if only as a collector's item. The item number is #200122

LOADSTAR 1-800-594-3370

The Internet for Commodore C64/128 Users

3rd Edition

by Gaelyne R. Gasson

ISBN:0-9585837-0-6

The only Commodore C64/128 Internet reference guide, this 300+ page manual takes you through hardware and software needed, how to get online and what you can do once you're there. It covers Email, World Wide Web, FTP, IRC, Telnet, Newsgroups, Commodore files, archives and much more.

ONLY $29.95 US + $7.00 shipping via Economy Airmail

Visa, MasterCard, Amex and personal checks welcome.

VideoCam Services

90 Hilliers Rd, Reynella 5161, Sth Australia

Phone: +61 8 8322-2716 **Email: videocam@videocam.net.au**
Fax: +61 8 8387-5810 **WWW: http://videocam.net.au**

Also available from Loadstar. Item #900920

The Blarney Stone — 403-246-5290

Calgary, Alberta
BBS Software/Networks....C-Net 128/CommNet
BBS Platform/Hardware....C128/LtK
System Operator Name.....Scott McGowan
System Operator Handle...Irish
Maximum Modem Speed.....2400 BPS
Open Status Verified.....4 Jun 1998 - Ron Fick

Warp Nine — 905-427-6798

Ajax, Ontario
BBS Software/Networks....C*Base 64 v3.0
BBS Platform/Hardware...C64
(scotty_78@hotmail.com)
System Operator Handle...Scotty
Maximum Modem Speed.....2400 BPS
Open Status Verified.....27 Oct 1997 - System Operator

Hyperspace — 403-274-0771

Location.....Calgary, Alberta
BBS Software/Networks....Omni 128
BBS Platform/Hardware....C128/RL CHD
System Operator Name.....Mike Stoll (stollm@cadvision.com)
System Operator Handle...Zaphod
Maximum Modem Speed.....14.4K BPS
Open Status Verified.....4 Jun 1998 - Ron Fick
Club BBS for CCUG

Robyn's Nest — 905-579-9547

Oshawa, Ontario
BBS Software/Networks....C-Net 64 DS2/CommNet
BBS Platform/Hardware....C64
System Operator Name.....Not Given (alm-pl@speedline.ca)
System Operator Handle...Robyn
Maximum Modem Speed.....2400 BPS
Open Status Verified On..4 Jun 1998 - System Operator

Kapital K'pers — 941-656-5613

N. Fort Myers, Florida
BBS Software/Networks....Image/CommNet
BBS Platform/Hardware....C64/SC64
System Operator Name.....Jim Caldwell (kapital1@juno.com)
System Operator Handle...Range Rover
Maximum Modem Speed.....14.4K BPS
Open Status Verified.....11 Sep 1998 - System Operator
Gateway from Image to DS2

Northern Outpost — 403-622-3395

Fox Creek, Alberta
BBS Software/Networks....C-Net 128/CommNet
BBS Platform/Hardware....C128/LtK
System Operator Name.....Tom Gislason (gislason@telusplanet.net)
System Operator Handle...Quinn the Eskimo
Maximum Modem Speed.....2400 BPS
Open Status Verified.....4 Jun 1998 - Ron Fick

The Deadworld — 905-720-3323

Oshawa, Ontario
BBS Software/Networks....C-Net 64 DS2/CommNet
BBS Platform/Hardware....C64
System Operator Name.....Paul Van Dolewcerd
System Operator Handle...Paul Van Dolewcerd
Maximum Modem Speed.....2400 BPS
Open Status Verified.....4 Jun 1998 - Ron Fick

Star Base 2 — 941-748-6618

Bradenton, Florida
BBS Software/Networks....Color 64
BBS Platform/Hardware....C64/SC64
Maximum Modem Speed.....2400 BPS
Open Status Verified.....25 Oct 1997 - Kenneth Nealey

The Vault — 416-694-2193

Toronto, Ontario
BBS Software/Networks....Centipede (formerly RAW)/ComLink,CommNet,Net64
BBS Platform/Hardware....C128/JD - 64K VDC, 1.2 GB CHD, CBM 1750 REU, Boca 14.4K Modem
System Operator Name.....Mark Wigston (thevault@mypad.com & strat@mypad.com)
System Operator Handle...***STRAT***
Maximum Modem Speed.....14.4K BPS
Open Status Verified.....4 Jun 1998 - Ron Fick
In it for the long haul! Don't let the dream die.

Credits

Kenneth Nealey.....(wizard_2@pacbell.net).....25 Oct 97
Fungus - F4CG/Carcass...(fungus@c64.org).....30 Oct 97
Ed Paulsen.....(epaulsen@netsle.net).....20 Nov 97
Ron Fick.....(rfick@myNX.net).....04 Jun 98
Pat Keller, Jr.....(pkeller1@ibm.net).....19 Jul 98
Heman Vergara.....(hvergara@coyote.accessnv.com)...02 Sep 98
Jim Caldwell.....(jim.caldwell@neverending.com)...10 Sep 98
And to all of the system operators who provided their information!

LOADSTAR LETTER SUBSCRIPTION

Re-subscribe To The LOADSTAR Letter. Don't miss a single Typo. Give your friends and even your enemies a gift subscription!

Send Coupon To:
LOADSTAR Letter
606 Common Street
Shreveport LA 71101

Name

Address:

☐ Here's $18 to renew my subscription! International subscribers remit $20 in US funds

Credit Card _____ Account # _____

Exp. Date _____ Signature _____

Ridiculous Addresses

•dam.mit.edu
•monarch.butterfly.net
•gratuitouslylonghostname.apana.org.au
•drag.net
•my-hostname-is-longer-than-yours.mit.edu
•tragically.hip.berkeley.edu
•dislocated.hip.berkeley.edu
•ohsaycan.ucc.american.edu
•huh_huh.fire.com
•vo.mit.edu

BBS Commandments

1. Thou shall love thy BBS with all thy heart and all thy bytes.
2. Thou shalt remember thy name and password.
3. Thou shalt only call a BBS two times a day.
4. Honor thy SysOp.
5. Thou shalt not covet thy neighbor's password, nor his or her real name, computer, software, nor any other thing belonging to him or her.
6. Thou shalt not post messages that are stupid, worthless, or have no meaning.
7. Thou shalt use the English language properly.
8. Thou shalt spell thy words correctly whenever possible.
9. Thou shalt delete thine olden messages.
10. Thou shalt help other users.
11. Thou shalt not post anonymously when offering criticism.
12. Thou shalt keep thy foul language to thyself.
13. Woe be unto the user who attempt to crash thy BBS, for he or she shalt be cast out from the sanctuary of thy hobby and must repent by doing 40 days and 40 nights of penance of voice-only communications.
14. Thou shalt first dial BBS numbers during the day by way of voice line to assure correct numbers.
15. Thou shalt not post messages while drunk.
16. Thou shalt confine thy messages to those of friendship, requests for assistance, aid to the needy, advice, and advancement of thy hobby; and thou art obligated to repel any who wouldst transgress upon those commandments.
17. If thou doth promise to reply to a message and thou doth not, then surely thou shalt spill coffee into thy keyboard and burn out thy central processing chip.
18. Thou shalt not giveth any false information when applying for membership to a BBS, for verily it is written that whosoever shall do so will surely be found out and thy welcome on all boards will be thus denied forever and ever.
19. Thou shalt log on properly and in accordance with the SysOp's rules.
20. Thou shalt observe BBS time limits.
21. Thou shalt not upload "worm" programs.
22. Thou shalt not ask stupid questions that are already fully explained in the BBS instructions.
23. Thou shalt not exchange copy protected software thru the BBS.
24. Thou shalt not violate applicable state/ federal/local laws hand regulations affecting BBS telecommunications, or thy will feel the wrath of thy judicial system.
25. Thou shalt not hack.

LOADSTAR LETTER #61

J&F PUBLISHING 606 COMMON STREET SHREVEPORT LA 71101

- COMMODORE NEWS
- COMMODORE VIEWS

Bulk Rate
U.S Postage PAID
Shreveport LA
PERMIT #85